

云南大学数学系《运筹学通论实验》课程上机实验报告

课程名称: 运筹学通论	学期: 2015-2016 学年第二学期	成绩:
指导教师: 李建平	学生姓名: 金洋	学生学号: 20131910023
实验名称: Min&IndexesOfMin&Set		
实验编号: No.1	实验日期: 2016/3/4	实验学时: 1
学院: 数学与统计学院	专业: 信息与计算科学	年级: 2013

一、实验目的

使用 c 语言实现求一实序列的最小值及相应的所有下标, 并将所有最小值下标保存于一集合;

二、实验内容

给定两组实序列 a_1, a_2, \dots, a_n 及 b_1, b_2, \dots, b_n , 求序列 c_1, c_2, \dots, c_n 的最小值及相应

的所有下标, 其中 $c_i = \begin{cases} \frac{a_i}{b_i}, & \text{if } b_i \neq 0 \\ +\infty, & \text{otherwise} \end{cases}$

三、使用环境

平台: Microsoft Visual C++ 6.0

语言: C 语言

四、算法介绍

Algorithm Min&IndexesOfMin&Set

Input n, a[i], b[i], (i=1, 2, ..., n);

Output the original data, min, the indexes of the minimum;

Begin

Step 1: For i=1 through n

 If (b[i]==0) then c[i]=+∞

 else c[i]=a[i]/b[i];

Step 2: $\text{min} = +\infty$, $\text{indexOfMin} = 0$;

Step 3: For $i=1$ through n

 If ($c[i] < \text{min}$) then

 1. $\text{min} = c[i]$;

 2. $\text{indexOfMin} = i$;

Step 4: $\text{indexNum} = 1$; $\text{index}[1] = \text{indexOfMin}$;

Step 5: For $i = \text{indexOfMin} + 1$ through n

 If ($c[i] == \text{min}$) then

 1. $\text{indexNum} = \text{indexNum} + 1$;

 2. $\text{index}[\text{indexNum}] = i$;

End.

五、调试过程

1. 程序代码

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define MAXNUM 10000
```

```
#define MAX 1000000
```

```
void main() {
```

```
    float a[MAXNUM], b[MAXNUM], c[MAXNUM], min;
```

```
    int i, indexOfMin, n, indexNum, index[MAXNUM];
```

```
    /*Input*/
```

```
    printf("Please input n:");
```

```
    scanf("%d", &n);
```

```
    printf("Please enter the array a:\n");
```

```
for(i=1;i<=n;i++)
    scanf("%f",&a[i]);

printf("Please enter the array b:\n");

for(i=1;i<=n;i++)
    scanf("%f",&b[i]);

/*Calculate*/

for(i=1;i<=n;i++) {
    if ( fabs(b[i])<1e-7 ) //判断 b[i]==0? 考虑到浮点类型的精度问题, 使用绝对值与 10^-7 比较
        c[i]=MAX;
    else
        c[i]=a[i]/b[i];
}

min=MAX;
indexOfMin=0;
for(i=1;i<=n;i++) {
    if (c[i]<min) {
        min=c[i];
        indexOfMin=i;
    }
}

indexNum=1;
index[indexNum]=indexOfMin;
```

```
for(i=indexOfMin+1;i<=n;i++) {  
    if (c[i]==min) {  
        indexNum++;  
        index[indexNum]=i;  
    }  
}  
  
/*Output*/  
printf("The original data:a,b,c\n");  
for (i=1;i<=n;i++) {  
    printf("%3d:%16.2f,%16.2f,%16.2f\n",i,a[i],b[i],c[i]);//输出结果保留两位  
    小数  
}  
printf("\nThe minimum is %16.2f\n",min);  
printf("The indexes of minimum: ");  
for (i=1;i<=indexNum;i++)  
    printf("%d ",index[i]);  
printf("\n");  
}
```

2. 运行窗口

```
"G:\O.R\实验1-Min&IndexOfMin\Debug\Min&IndexesOfMin&Set.exe"
Please input n:12
Please enter the array a:
0 0 1 2 3 4 5 8 7 10.365 -100 -98
Please enter the array b:
7 8 9 5 4 1 5 4 7 4 50 49
The original data:a,b,c
  1:      0.00,      7.00,      0.00
  2:      0.00,      8.00,      0.00
  3:      1.00,      9.00,      0.11
  4:      2.00,      5.00,      0.40
  5:      3.00,      4.00,      0.75
  6:      4.00,      1.00,      4.00
  7:      5.00,      5.00,      1.00
  8:      8.00,      4.00,      2.00
  9:      7.00,      7.00,      1.00
 10:     10.36,      4.00,      2.59
 11:    -100.00,     50.00,     -2.00
 12:     -98.00,     49.00,     -2.00

The minimum is      -2.00
The indexes of minimum: 11 12
Press any key to continue
```

```
"G:\O.R\实验1-Min&IndexOfMin\Debug\Min&IndexesOfMin&Set.exe"
Please input n:3
Please enter the array a:
1 2 3.69
Please enter the array b:
0 0 -9.65895
The original data:a,b,c
  1:      1.00,      0.00,     1000000.00
  2:      2.00,      0.00,     1000000.00
  3:      3.69,     -9.66,      -0.38

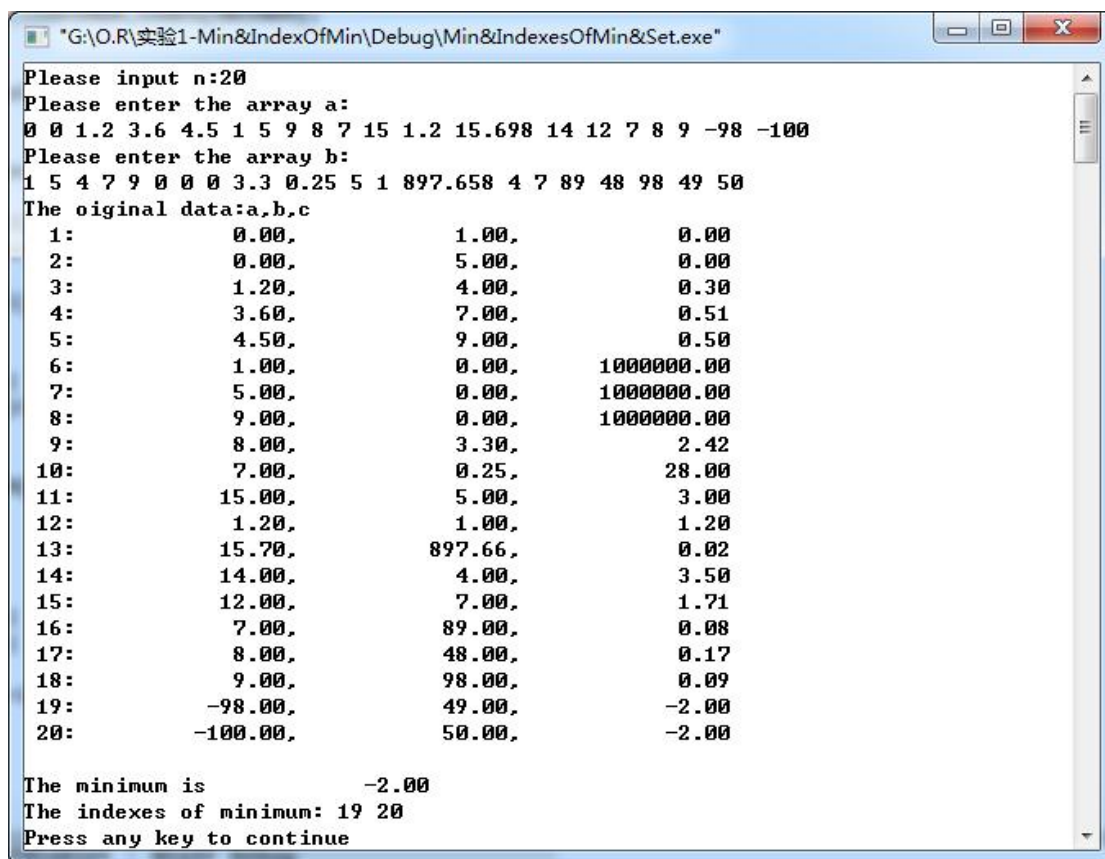
The minimum is      -0.38
The indexes of minimum: 3
Press any key to continue
```

```

G:\O.R\实验1-Min&IndexOfMin\Debug\Min&IndexesOfMin&Set.exe
Please input n:25
Please enter the array a:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 -21 -22 -23 -24 -25
Please enter the array b:
-0.1 -0.2 -0.3 -0.4 -0.5 -0.6 -0.7 -0.8 -0.9 -1.0 -1.1 -1.2 -1.3 -1.4 -1.5 -1.6
-1.7 -1.8 -1.9 -2.0 2.1 2.2 2.3 2.4 2.5
The original data:a,b,c
1:      1.00,      -0.10,      -10.00
2:      2.00,      -0.20,      -10.00
3:      3.00,      -0.30,      -10.00
4:      4.00,      -0.40,      -10.00
5:      5.00,      -0.50,      -10.00
6:      6.00,      -0.60,      -10.00
7:      7.00,      -0.70,      -10.00
8:      8.00,      -0.80,      -10.00
9:      9.00,      -0.90,      -10.00
10:     10.00,     -1.00,      -10.00
11:     11.00,     -1.10,      -10.00
12:     12.00,     -1.20,      -10.00
13:     13.00,     -1.30,      -10.00
14:     14.00,     -1.40,      -10.00
15:     15.00,     -1.50,      -10.00
16:     16.00,     -1.60,      -10.00
17:     17.00,     -1.70,      -10.00
18:     18.00,     -1.80,      -10.00
19:     19.00,     -1.90,      -10.00
20:     20.00,     -2.00,      -10.00
21:    -21.00,      2.10,      -10.00
22:    -22.00,      2.20,      -10.00
23:    -23.00,      2.30,      -10.00
24:    -24.00,      2.40,      -10.00
25:    -25.00,      2.50,      -10.00

The minimum is      -10.00
The indexes of minimum: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25
Press any key to continue

```



```

G:\O.R\实验1-Min&IndexOfMin\Debug\Min&IndexesOfMin&Set.exe
Please input n:20
Please enter the array a:
0 0 1.2 3.6 4.5 1 5 9 8 7 15 1.2 15.698 14 12 7 8 9 -98 -100
Please enter the array b:
1 5 4 7 9 0 0 0 3.3 0.25 5 1 897.658 4 7 89 48 98 49 50
The original data:a,b,c
  1:      0.00,      1.00,      0.00
  2:      0.00,      5.00,      0.00
  3:      1.20,      4.00,      0.30
  4:      3.60,      7.00,      0.51
  5:      4.50,      9.00,      0.50
  6:      1.00,      0.00,    1000000.00
  7:      5.00,      0.00,    1000000.00
  8:      9.00,      0.00,    1000000.00
  9:      8.00,      3.30,      2.42
 10:      7.00,      0.25,     28.00
 11:     15.00,      5.00,      3.00
 12:      1.20,      1.00,      1.20
 13:     15.70,    897.66,      0.02
 14:     14.00,      4.00,      3.50
 15:     12.00,      7.00,      1.71
 16:      7.00,    89.00,      0.08
 17:      8.00,    48.00,      0.17
 18:      9.00,    98.00,      0.09
 19:     -98.00,    49.00,     -2.00
 20:    -100.00,    50.00,     -2.00

The minimum is      -2.00
The indexes of minimum: 19 20
Press any key to continue
  
```

六、总结

- 1.学会使用 c 语言判断浮点数据类型是否为 0;
- 2.学会使用 c 语言实现求一实序列的最小值及相应的所有下标;

七、参考文献

- [1] 谭浩强著,《c 程序设计》(第三版),清华大学出版社,2005.7;
- [2] 《运筹学》教程编写组,《运筹学》(第 4 版),清华大学出版社,2013.1;

八、教师评语